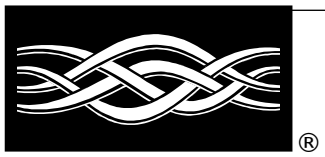
Microsoft®

Microsoft® Windows NT® Server

Server Operating System



White Paper

Microsoft Windows NT Active Directory:
An Introduction to the Next Generation Directory Services

© 1997 Microsoft Corporation. All rights reserved.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft, BackOffice, the BackOffice logo, Visual Basic, Win32, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Java is a trademark of Sun Microsystems, Inc. Other product and company names may be the trademarks of their respective owners.

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399 USA

0797

Microsoft® Windows NT® Server

Abstract

This white paper provides an introduction to the Active Directory, Microsoft's next generation directory services for Microsoft® Windows NT® Server. The Active Directory uses the Internet concept of a *name space* to provide a single point of administration and replication, a hierarchical view of the directory, extensibility, scalability, distributed security, and multimaster replication.

This paper examines the key features and components of the Active Directory, and explains in some detail the effects these services will have on computing in the enterprise.

CONTENTS

Introduction....21
The Next Generation Directory Services 21
Beyond the Traditional Directory 21
The Directory as a Service Provider 2
Summary of Features and Benefits 23

Active Directory Overview....24
A Unified Directory 24
A Single Point of Administration 24
Scalability 24
Operating System Integration 25

Support for Open Standards and Standard Name Formats....26
Support for DNS 26
Creating Internet-Ready Names 26
Creating an Easier to Manage DNS Environment 27
Support for LDAP and HTTP 28
Support for Standard Name Formats 28
RFC822 Names 29
HTTP URL Names 29
LDAP URLs and X.500 Names 29
UNC Names 29

Application Programming Interfaces....210
Active Directory Service Interfaces (ADSI) 210
Windows Messaging API (MAPI) 211
LDAP C API 211

Enabling Massive Scalability....212
Using Domain Trees 212
Using the Container Hierarchy to Model the Organization 213
Gaining Finer-Grained Administration 213
Increasing Security 214
Extending the Directory Services via an Extensible Schema 214
The Global Catalog 214
Multimaster Replication 215
Updating the Directory in a Multimaster Environment 215
Support for Volatile Objects and Properties 216

Distributed Security....217
Private Key Security 217
Public Key Security 217

Administration Tools....218

INTRODUCTION

Drag-and-Drop Administration 218
Scripting and OLE Automation 218

Assuring Backward Compatibility 219
Easy Migration from Windows NT 3.5*x* and 4.0 219
Migrating from Windows NT 4.0 to the Active Directory 219
Easy Migration from the Microsoft Exchange Directory 221

Summary 222
For More Information 222

If you work in a large corporation today and need to find a certain file, but you aren't sure exactly where it is stored or even which server on your network holds the file, what can you use to see your network's directory structure? How can you "drill down" on each server if you don't even know all the server names? If you manage an enterprise network and you have a set of internal servers for your users, servers for sensitive data, an intranet with both restricted and open areas, and a public Internet site, how can you see everything you have from a single view? How can you set restrictions so that others do not have access to sensitive data? How can you replicate data so that it is controlled, yet available? And most importantly: how can you ensure that your directory solution will function across network and operating system boundaries and will grow as your network grows?

Today's Microsoft® Windows NT® Server network operating system offers the Windows NT Directory Services, a robust directory that delivers what customers need most—a single network logon and a single point of administration and replication. While these functions are critical to businesses, it is becoming increasingly clear that Windows NT Server enterprise customers need and want more from their directory services. They demand features such as a hierarchical view of the directory, extensibility, scalability, distributed security, and multimaster replication. To meet these needs, Microsoft is developing the Active Directory services, scheduled to be fully implemented in the next release of Windows NT Server.

The Next Generation Directory Services

The need for an ever more powerful, transparent, and tightly integrated directory system is driven by the explosive growth of networked computing. As LANs and WANs grow larger and more complex, as networks are connected to the Internet, and as applications require more from the network and are linked to other systems through corporate intranets, more is required from a directory service.

The protocols and object formats that a directory supports are the measure of its *openness*—that is the degree to which the directory is available to clients beyond those explicitly designed to use it. The application programming interfaces (APIs) supported define the range of tools and applications that will directly take advantage of the directory service. And the directory services must become a point of unification, providing a sense of order and structure, especially when managing information from competing network operating system (NOS) and application directories.

The Active Directory supports a wide range of well-defined protocols and formats and provides powerful, flexible, and easy-to-use APIs. Moreover, the Active Directory provides administrators and users with a one-stop source for resource and management information.

Beyond the Traditional Directory

Traditionally, directory services have been tools for organizing, managing, and locating “interesting” objects in a computing system. “Interesting” objects are things users (and applications) need to do their jobs: such as printers, documents, e-mail addresses, databases, users, distributed components, and other resources.

In their simplest form, directory services are like the white pages of a telephone book Using specific input (a person’s name, for example), a user can receive specific output (a person’s address and telephone number). Directory services also provide the functionality of the yellow pages Using general input (e.g., “where are the printers?”), a user can receive a listing of printer resources that can be browsed.

But directory services must do more as networked environments become larger and more complex, even before connecting to the global environment of the Internet. The Active Directory was created to meet the challenge of unifying and bringing order to diverse server hierarchies, or name spaces.

The Directory as a Service Provider

In addition to handling the traditional administrative tasks of the directory services, the Active Directory will satisfy a wide variety of naming, query, administrative, registration, and resolution needs. The following diagram summarizes its overall function in the system.

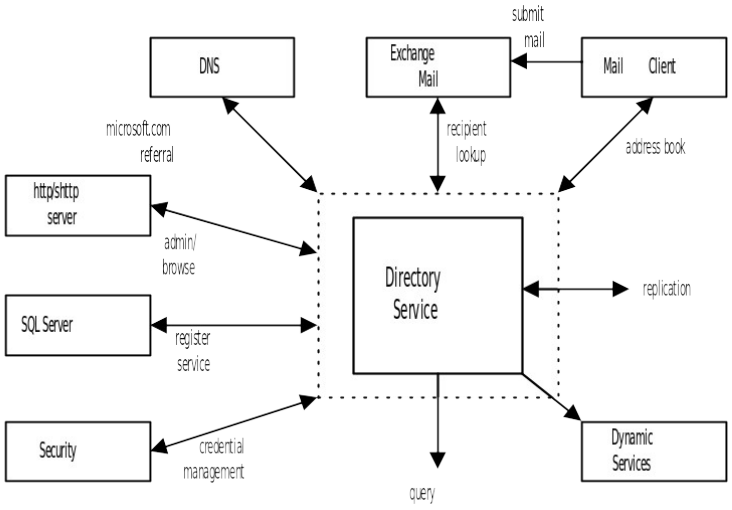


Figure 1. The directory is a service provider used to locate all network services and information.

The architecture of the Active Directory allows it to scale from the smallest of businesses to enterprises supporting international corporations and entire government departments and services.

The Active Directory uses a tightly integrated set of APIs and protocols to extend its services across multiple name spaces, and to gather and present directory and resource information that resides on different operating systems and at distant locations. For example, Microsoft today provides a rich set of interoperability components for Novell NetWare 3.*x*/4.*x* customers. As protocols evolve, Microsoft will work with the industry to standardize communications with other environments as well. To make the migration to Active Directory services as painless as possible, Microsoft will also ensure that earlier releases of Windows NT Server interoperate with later releases.

Summary of Features and Benefits

The Active Directory includes the following features and benefits:

- Support for open standards to facilitate cross-platform directory services, including support for the Domain Name System (DNS) and support for standard protocols, such as LDAP ~~and HTTP~~.
- Support for standard name formats to ensure ease of migration and ease of use.
- A rich set of APIs, which are easy to use for both the scripter and C/C++ programmer.
- Simple, intuitive administration through a simple hierarchical domain structure and the use of drag-and-drop administration.
- Directory object extensibility via an extensible schema.
- Fast lookup ~~and Internet publishing~~ via the global catalog.
- Speedy, convenient updates through multimaster replication.
- ~~Support for services that have a short life span such as chat~~

services, IP telephony, as well as other conferencing services.

- Backward compatibility with previous versions of the Windows NT operating system.
- Interoperability with NetWare environments.

The remainder of this document explores the design goals and implementation of the Active Directory for Windows NT Server 5.0, and explains how an enterprise can prepare for and ensure a seamless migration to the next generation of directory services.

The Active Directory is a directory service that is completely integrated with Windows NT Server and offers the hierarchical view, extensibility, scalability, and distributed security required by all business customers. For the first time, network administrators, developers, and users gain access to a directory service that:

- is seamlessly integrated with both Internet and intranet environments.
- provides simple, intuitive naming for the objects it contains.
- scales from a small business to the largest enterprise.
- works with familiar tools, such as Web browsers.
- provides simple, powerful, open application programming interfaces.

The Active Directory is a critical part of the distributed system. It allows administrators and users to use the directory service as a source of information, as well as an administrative service.

A Unified Directory

The Active Directory integrates the Internet concept of a *name space* with the operating system's directory services, thus allowing enterprises to unify and manage the multiple name spaces that now exist in the heterogeneous software and hardware environments of corporate networks. It uses the Lightweight Directory Access Protocol (LDAP) as its core protocol and can work across operating system boundaries, integrating multiple name spaces. It can subsume and manage application-specific directories, as well as other NOS-based directories, to provide a general-purpose directory that can reduce the administrative burden and costs associated with maintaining multiple name spaces.

The Active Directory is not an X.500 directory. Instead, it uses LDAP as the access protocol and supports the X.500 information model without requiring systems to host the entire X.500 overhead. The result is the high level of interoperability required for administering real-world, heterogeneous networks.

A Single Point of Administration

The Active Directory allows a single point of administration for all published resources, which can include files, peripheral devices, host connections, databases, Web access, users, other arbitrary objects, services, and so forth. It uses the Internet Domain~~irectory~~ Name Service~~pace~~ (DNS) as its locator service, organizes objects in domains into a hierarchy of organizational units (OUs), and allows multiple domains to be connected into a tree structure. Administration is further simplified because there is no notion of a primary domain controller (PDC) or backup domain controller (BDC). The Active Directory uses domain controllers (DCs) only, and all DCs are peers. An administrator can make changes to any DC, and the updates will be replicated on all other DCs.

Scalability

The Microsoft Exchange 4.0 directory structure and storage engine provide the foundation for the Active Directory. The Microsoft Exchange storage engine provides multiple indexes for fast retrieval and an efficient mechanism for storing “sparse” objects. That is, objects that support many different properties but do not always have values for all of them. From this foundation, Microsoft has developed general-purpose directory services that scale from a small installation with a few hundred to a few thousand objects, to a very large installation with millions of objects.

The Active Directory supports multiple stores and can hold more than 1~~0~~ million objects per store, thus offering unparalleled scalability while maintaining a simple hierarchical structure and ease of administration. When combined with the Microsoft Distributed File System (scheduled for release with Windows NT Server 5.0), the Active Directory will bring networks even closer to the goal of a single global name space.

Operating System Integration

The Active Directory is seamlessly integrated with Windows NT Server, which is the only operating system that offers traditional file and print, applications, communications, and Internet/intranet support built into the base product. Windows NT Server is the best file and print server for all of a business’s information and resource sharing needs, outperforming all other operating systems available today. It is also the best applications server available, offering the best scalability/price ratio in the industry. Additionally, Windows NT Server is an excellent communications platform, offering such features as Remote Access Services, TAPI, and PPTP.

SUPPORT FOR OPEN STANDARDS AND STANDARD NAME FORMATS

The overriding goal of the Active Directory is to provide a unified view of the network that will greatly reduce the number of directories and name spaces with which network administrators (and users) must contend. The Active Directory is specifically designed to subsume and manage other directories, regardless of their location or their underlying operating system(s). To accomplish this, the Active Directory provides extensive support for existing standards and protocols, including standard name formats, and provides application programming interfaces (APIs) that facilitate communication with these other directories.

The Active Directory uses the Domain Name System (DNS) as its ~~name system~~location service, and can exchange information with any application or directory that uses Lightweight Directory Access Protocol (LDAP)~~ or Hypertext Transfer Protocol (HTTP)~~,

Support for DNS

The Active Directory combines the best of DNS as a locator service with the best of X.500, while avoiding the failings of both *and* advancing Internet standards.

DNS is the most widely used directory service in the world. DNS is the locator service used on the Internet and in most private intranets. A locator service is used to translate a name—for example, **MyMachine**.**Myco.Com**—into a TCP/IP address. DNS is designed to scale to very large systems (it supports the entire Internet), while remaining “lightweight” enough for use in a system with just a few computers.

Creating Internet-Ready Names

The Active Directory also uses DNS as its locator service. In the Active Directory, Windows NT Domain Names *are* DNS names. Users will find the same simple naming used on the Internet in the Active Directory. Myco.Com can be both a DNS domain (that is, an area of addressing) and a Windows NT Domain. **JamesSmith@Myco.Com** is both an Internet e-mail address and a user name in the Myco.Com domain. Windows NT domains can be located on the internet and Intranet the same way any resource is located on the Internet—by means of DNS. This is shown in Figure 2.

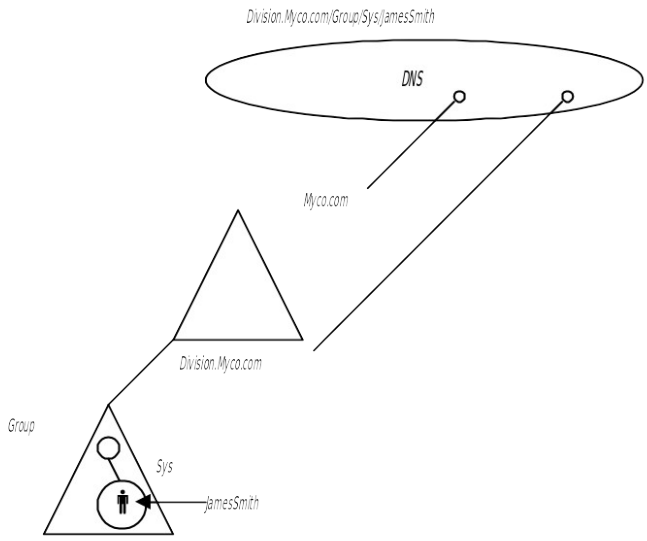


Figure 2. DNS is the Windows NT Locator Service.

Creating an Easier to Manage DNS Environment

DNS has historically been somewhat difficult to manage because it required the manual maintenance of text files containing the friendly name-to-address mapping for every computer in an organization. In Windows NT Server version 4.0, Microsoft introduced a DNS Server with built-in Windows® Internet Naming Service integration. The DNS Server in Windows NT includes a graphical administration tool designed to make editing DNS files less cumbersome. To eliminate manual assignment of addresses, the DNS Server in Windows NT is tightly integrated with the Windows Internet Naming Service, a Windows NT service that dynamically updates the friendly name-to-address mapping file.

In a Windows NT Server-based network, client computers are automatically assigned TCP/IP addresses at startup using the Dynamic Host Configuration Protocol (DHCP). The clients then register their names and addresses in the Windows Internet Naming Service. This is shown in Figure 3.

The DNS Server in Windows NT uses the Windows Internet Naming Service to resolve unrecognized names in the organization it serves.

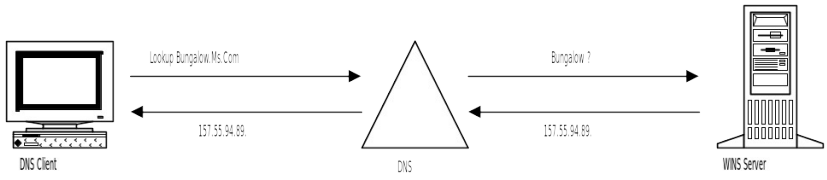


Figure 3. DNS and WINS are integrated to provide dynamic DNS updates.

Microsoft’s solution for dynamically updating DNS tables—integrating

DNS and the Windows Internet Naming Service—is a short-term solution. ~~Currently, t~~The Internet standards for DNS have been~~are being~~ updated to support *Dynamic DNS*. Dynamic DNS eliminates the need for WINS because it allows clients with dynamically assigned addresses to register directly with the DNS server and update the DNS table on the fly. Servers running the Active Directory will use Dynamic DNS to publish themselves in DNS. By deploying Windows NT 4.0, DNS, and WINS today, systems administrators create the foundation for the Active Directory and Dynamic DNS.

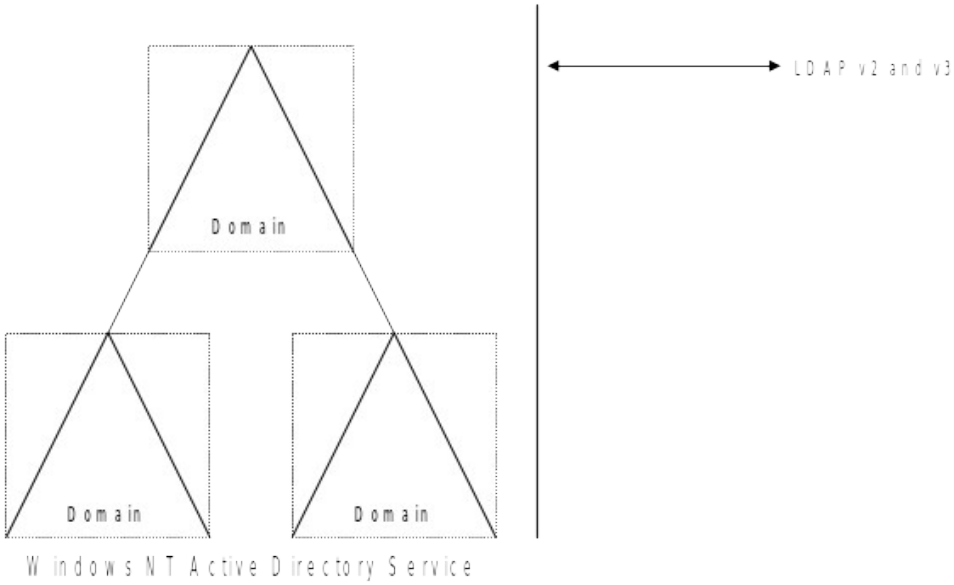


Figure 4. Active Directory supports the LDAP v2 and v3~~and HTTP~~ protocols.

Support for LDAP~~ and HTTP~~

The Active Directory further embraces Internet standards by directly supporting the Lightweight Directory Access Protocol (LDAP) ~~and the Hypertext Transfer Protocol (HTTP)~~.

LDAP is a~~n~~ Internet Proposed ~~s~~Standard (RFC2251~~1777~~) for accessing directory services, and was developed as a simpler alternative to the X.500 DAP protocol. Microsoft is an active participant in the advancement of LDAP standards, and provides support for both LDAP version 2 and version 3 (~~currently in draft form)~~ in the Active Directory.

~~HTTP is the standard protocol for displaying pages on the World Wide Web. Every object in the Active Directory can be displayed as an HTML page in a Web browser. Directory support extensions to the Microsoft Internet Information Server (IIS) translate HTTP requests for directory objects into HTML pages for viewing in any HTML client. Thus, users receive the benefit of the familiar Web browsing model when querying and viewing objects in the Active Directory.~~

Support for Standard Name Formats

Both users and applications are affected by the name format used in directory services. If a user or application needs to find or use something, that user or application must know the name or some property of the object in order to locate it. There are several common forms for names in directories, defined by both formal and de facto standards, and the Active Directory supports many of these formats. This extended support for diverse name formats allows users and applications to use the format that they are most familiar with when accessing the Active Directory. Some of these formats are explained next.

RFC822 Names

RFC822 names are in the form ***somename@somedomain*** and are familiar to most users as Internet e-mail addresses; e.g., **JamesSmith@myco.Com**. The Active Directory provides a "friendly name" in RFC822 form for all users~~objects~~. For example, a user can use a friendly name as an e-mail address, suitable for display on a business card, *and* as the name used to log on.

~~HTTP URL Names~~

~~The Active Directory supports access from Web browsers via the HTTP protocol and Microsoft Internet Information Server. HTTP uniform resource locators (URLs) are familiar to most users who have Web browsers, and are in the form **HTTP://*somedomain/path-to-page***.~~

~~The Active Directory supports access to its contents via HTTP URLs in which ***somedomain*** refers to a server running Active Directory services and ***path-to-page*** is the path through the Active Directory hierarchy to the object of interest, for example:~~

```
~~HTTP://SomeServer.Myco.Com/BIN/Division/Product/Ssys/JamesSmith~~
```

LDAP URLs and X.500 Names

The Active Directory supports access via the LDAP protocol from any LDAP-enabled client. LDAP names are less intuitive than Internet names, but the complexity of LDAP naming is usually hidden within an application. LDAP names use the X.500 naming convention called *attributed naming*. An LDAP URL names the server holding Active Directory services and the attributed name of the object, for example:

```
LDAP://SomeServer.Myco.Com/CN=jamessmith,OU=Sys,OU=Product,
OU=Division,~~0~~DC=myco,DC=Com~~C=US~~
```

~~UNC Names~~

~~The Active Directory supports the Universal Naming Convention (UNC) used in Windows NT Server-based networks to refer to shared volumes, printers, and files. A user can refer to a shared file published in the Active Directory by a UNC name, for example:~~

APPLICATION PROGRAMMING INTERFACES

\\Myco.Com\division.product.Sys.SomeVolume\XLSheets\Budget.XLS

The Active Directory provides powerful, flexible, and easy-to-use application programming interfaces (APIs). The availability of a rich set of APIs for the directory service encourages the development of applications and tools that make use of the directory's services. The Active Directory includes three major API sets:

- **ADSI,** the Active Directory Service Interfaces, a set of component object model (COM) interfaces for manipulating and querying multiple directory services.
- **MAPI**, the Windows® Open Services Architecture Messaging API.
- **LDAP C API** (RFC1823), an informational RFC that is the de facto standard in C programming for LDAP applications.

Each of these APIs is described next.

Active Directory Service Interfaces (ADSI)

To make it easier to write directory-enabled applications that access the Active Directory and other LDAP-enabled directories, Microsoft developed Active Directory Service Interfaces (ADSI). ADSI is a set of extensible, easy-to-use programming interfaces that can be used to write applications to access and manage the following:

- Active Directory
- any LDAP-based directory
- other Directory Services in a customer's network, including NDS

ADSI is part of the Open Directory Services Interface (ODSI), the Windows® Open Services Architecture (WOSA) architecture for manipulating and querying multiple directory services. ADSI objects are available for Windows NT 4.*x,* Novell NetWare 3.*x,* and 4.*x,* and the Active Directory, as well as any other directory service that supports the LDAP protocol.

Active Directory Service Interfaces abstract the capabilities of directory services from different network providers to present a single set of directory service interfaces for managing network resources. This greatly simplifies the development of distributed applications, as well as the administration of distributed systems. Developers and administrators use this single set of directory service interfaces to enumerate and manage the resources in a directory service, no matter which network environment contains the resource. Thus, ADSI makes it easier to perform common administrative tasks, such as adding new users, managing printers, and locating resources throughout the distributed computing environment, and ADSI makes it easy for

developers to “directory enable” their applications.

ADSI is designed to meet the needs of traditional C and C++ programmers, system administrators, and sophisticated users. With ADSI, development of directory enabled applications is fast and easy. ADSI presents the directory as a set of COM objects, which provide behavior in addition to data. For example, an application can use an ADSI **PrintQueue** object to retrieve data, such as characteristics of the queue, and to pause the queue. In the Visual Basic® programming system, this is as easy as:

```
Dim MyQueue as IOleDsPrintQueue

set MyQueue =
GetObject("DS://Myco.Com/Division/Product/Printers/MyPrinter")

MyQueue.Pause
```

Because ADSI objects are available for many popular directory services, ADSI is an ideal tool for building applications that will work with multiple directories. In addition, a service provider writer may choose to supply rich query by supporting OLE DB interfaces. Thus, tools that take advantage of the OLE DB interfaces can use Active Directory service providers.

ADSI objects are designed to meet the needs of three main audiences:

- **Developers**. Typically, this audience will use ADSI with a compiled language such as C++, although Microsoft Visual Basic can be used for prototyping the application. For example, a developer could write an application to manage multiple directories, network printing, back up databases, and so on.
- **System administrators**. Typically, this audience will access ADSI through a scripting language, such as Microsoft Visual Basic, although C/C++ can also be used to enhance performance. For example, with Active Directory an administrator could write a script to add 100 new users to the system and establish them as members of selected security groups.
- **Users.** Like the system administrators, this audience will access ADSI through a scripting language. For example, a user might write a script to locate all print jobs in a group of print queues and display the status of each.

Windows Messaging API (MAPI)

The Active Directory provides support for MAPI so that legacy MAPI applications will continue to work with the Active Directory. Because of this, developers of new applications are encouraged to use ADSI to build their directory-enabled applications.

LDAP C API

The LDAP API provides a "lowest common denominator" solution for developers who need their applications to work on many different client types. Similarly, existing LDAP applications will run against Active Directory services with little or no modification beyond extending the application to support object types unique to the Active Directory. Developers of LDAP applications are encouraged to migrate to ADSI, which supports any LDAP-enabled directory services.

ENABLING MASSIVE SCALABILITY

Microsoft recognizes that not all businesses are the same size, and that there isn't much value in a directory in which the small business suffers at the low end and the large business suffers at the high end. This is why the Active Directory performs very well on just a single computer, and also scales to a large enterprise environment.

Windows NT 4.0 scales quite well up to at least 100,000 users, but the Active Directory can scale up to over 1 million~~s of~~ users ~~(10 million total objects)~~ in a single domain, and even larger numbers in a domain tree or Forest. The administrative granularity of the Active Directory allows small domains to be created that are easy to administer and that allow organizations, and the networks that support them, to grow very large. Large enterprises are not be administered any differently than smaller businesses—they just have more administrators.

The Active Directory scales by creating one copy of the directory store for each domain. This copy of the directory store holds the objects that apply to that domain only. If multiple domains are related, they can be built into a tree. Within this tree, each domain has its own copy of the directory store, with its own objects, and the ability to find all the other copies in the tree of the directory store.

Rather than creating a single copy of the directory that gets larger and larger, the Active Directory creates a tree made up of small pieces of the directory, each containing information that allows it to find all the other pieces. The Active Directory breaks the directory into pieces so that the part of the directory someone uses most often is closest to them. Other users in other locations may want to use that same part of the directory, and they would also have a copy close to them. All replicas of that part of the directory are kept synchronized. If a record in any copy is modified, the change is propagated to the other copy. This allows the Active Directory to scale up to many millions of users in a tree.

Using Domain Trees and Forests

The key to the scalability of Active Directory is the domain tree. Unlike directory services that consist of a single tree structure and require a complex "top down" partitioning process, the Active Directory provides a simple and intuitive "bottom up" method for building a large tree. In the Active Directory, a single domain is a complete partition of the directory. Domains are subdivided into organizational units (OUs) for administrative purposes. This can be seen in Figure 5.

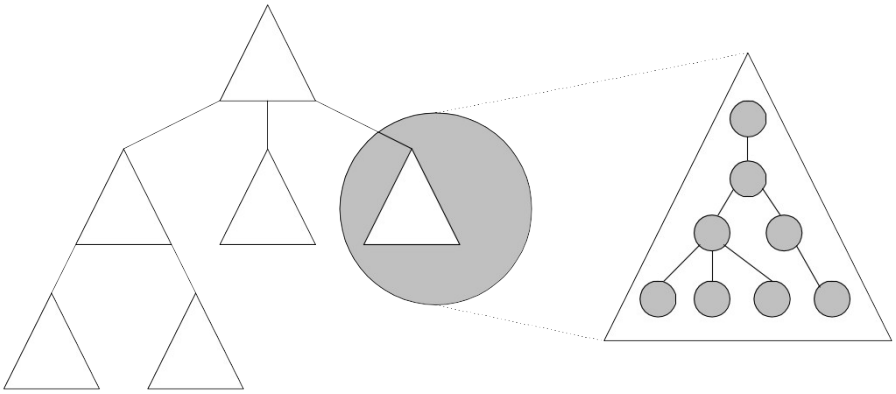


Figure 5. Active Directory uses domain trees and organizational units (OUs) to provide bottom up tree structures.

A single domain can start very small and grow to contain over 10 million objects. When a more complex organizational structure is required or a very large number of objects must be stored, multiple Windows NT domains can be easily joined together to form a tree. Multiple Trees can form a Forest. Forests allow an enterprise to include Windows NT Domains with disjoint DNS names, for example "MyCo.Com" and "MyCoResearch.Com".

Using the Container Hierarchy to Model the Organization

The ability of the container hierarchy in the Active Directory to nest organizational units within domains (as well as within other OUs) provides a hierarchical name space that administrators can use to reflect their organization and to delegate administrative control. A container contains a list of contents; for example, major company divisions. In this example, it is possible to select a division below a previous division and open it, and so on.

Moving to a container hierarchy with a finer-grained administrative model, such as the model used here, solves many problems. While large domains can still be used, finding things will be easy. Everything that exists in the domain tree will show up in the global catalog, a service that allows users to easily find an object, regardless of where it is in the tree or forest.

Gaining Finer-Grained Administration

The robust domain trees provided by the Active Directory offer far greater administrative flexibility than the single-tree organizational structures of other directory services. Although single-tree domains can be built with the Active Directory, a better administrative option is to build a tree of domains, each with its own security boundary. A hierarchy of domains allows for finer granularity of administration without compromising security. Permissions can flow down the tree, with users being granted permissions (as well as granting permissions to others) on an organizational unit basis. This domain-tree structure easily

accommodates organizational change with pruning, grafting, and merging.

Each domain in a domain tree has a copy of the directory service holding all objects for that domain and metadata about the domain tree, such as the schema, list of all domains in the tree, location of global catalog servers, and so forth. Since a single directory service store does not have to hold all objects for all domains, very large trees can be built without compromising performance.

Increasing Security

The Active Directory provides the fine-grained administration structure that allows for decentralized administration without compromising security. Because each domain is a security boundary, multiple security boundaries are possible. With this design administrators in domain A are not automatically administrators in domain B. The container hierarchy is important because, today, the scope of administration is the domain, and the administrator of a domain has authority over every object and service within that domain. The Active Directory grants privileges to users based on the specific functions they must perform within a given scope. Administrative scope can include an entire domain, a subtree of OUs within a domain, or a single OU.

With the Active Directory, very large structures of users can be created in which each user can potentially access all of the information stored in the directory, but the security boundaries remain clear. Security boundaries can also be much smaller than domains. For example, when a user account is created, it is associated with a particular domain, but it can also be put into an organizational unit. Permission to create users in an organizational unit can be delegated, allowing someone to create users or other directory objects in one place only, with rights within that OU only. In addition, OU hierarchies can be created. The Active Directory introduces many very specific permissions, all of which can be delegated and restricted in scope.

Extending the Directory Services via an Extensible Schema

To provide administrators with the power to create their own directory object types, the Active Directory is extensible through a schema mechanism. If a user has an important piece of information that the user wants to publish in the directory, he or she can create a whole new object type and publish it. For example, a wholesale distributor may want to create a warehouse object to put in its directory, with information that is specific to that business. New object classes can be defined and instances added.

The directory services themselves define a wide variety of classes. For example, the Active Directory provides standard objects for Domain, OU, User, Group, Machine, Volume, and PrintQueue, as well as a rich set of “connection point” objects used by Winsock, RPC, and DCOM services

to publish their binding information.

The Global Catalog

All objects stored in the Active Directory have entries in the global catalog (GC), a service that contains directory information from all of the source domains in the tree. Designed for high performance, the GC allows users to easily find an object—regardless of where it is in the tree—while searching by selected attributes. These attributes are contained in an abbreviated catalog. This technique, known *as partial replication*, allows many common queries to be resolved from the GC without requiring a lookup in the source domain.

The global view may contain any type of object, for example, Users, Services, or Machines. A typical use of the global view would be to provide a global address book for purposes of mail or any mail-enabled application.

Figure 6 shows the structure of the global catalog.

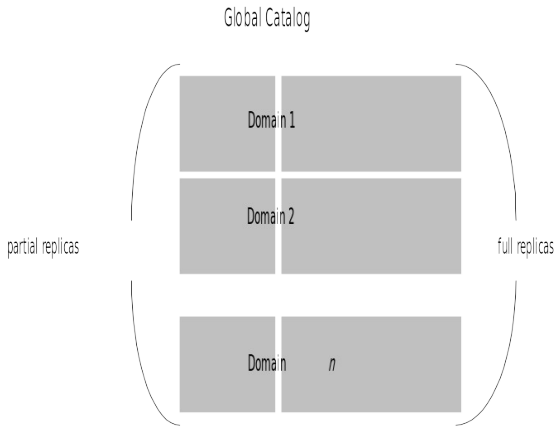


Figure 6. The global catalog structure provides access to full and partial replication.

Multimaster Replication

The manner in which a directory service stores information directly determines the performance and scalability of that directory service. Directory services must handle a very large number of queries compared to the number of updates. Typically the ratio is 99 percent query and 1 percent update. For this reason, replicated storage is important. By creating multiple replicas of the directory and keeping them consistent, the number of queries that can be handled with no performance degradation is increased. This reproduction and synchronization of directory information is known as *multimaster replication*.

Updating the Directory in a Multimaster Environment

The Active Directory offers true multimaster replication. Some directory

services use a *master-slave* approach to do updates: all of the updates must be made to the master copy of the directory, and these are then replicated to the slave copies. This is adequate for a directory with a small number of copies and an environment where all of the changes can be applied centrally, but this approach does not scale beyond small-sized organizations, nor does it address the needs of decentralized organizations. Because the Active Directory offers multimaster replication, individual changes made in one copy of the directory are automatically replicated to all other appropriate copies of the directory, whether connected via point-to-point or store-and-forward links.

Some directory services use time stamps to track updates. In a master-slave directory where all updates are made centrally, this is adequate, but in a multimaster replicating directory, time stamps are problematic. Unless time is perfectly synchronized among all copies of the directory, there is a chance for data loss or directory corruption. The Active Directory does not depend upon time stamps for detecting updates. Instead, it uses *Update Sequence Numbers* (USNs). Updates can be tracked because any time a user writes something into an object in the directory, it gets USN, which is held per computer and incremented any time a change is made to that object. If a user on one computer updates a user record, the current value for the update sequence number on that computer is incremented and then written into the object, along with the change and a unique signature of the first computer that wrote that change. The object also carries a USN for each property. When a property is updated, the new USN is advanced.

Changes are monitored, and the replication partners of one computer ask for all of its changes greater than the last USN received. The source computer will then search through the directory and find each object whose update sequence numbers are greater than the one presented by the partner machine.

Property changes are reconciled individually; when a change is replicated, only properties with a higher USN are updated. In the case of a collision (where two different computers have updated one property), the change with the later time stamp wins. The time stamp is used simply as an arbitrary “tie breaker,” so time synchronization is not important. Per-property reconciliation keeps the chance of collisions to a minimum.

Support for Volatile Objects and Properties

The architecture of the Active Directory supports the addition of objects and properties that are v*olatile*, that is, frequently changing or short-lived. This type of information is not usually stored in traditional directories because the information loses its accuracy before directory replication can propagate it. The Active Directory provides a mechanism to transparently link alternate information stores into the directories.

ADMINISTRATION
SECURITY

Volatile objects and properties are stored in separate storage with different replication characteristics while preserving a common user view of all objects, both static and volatile.

Private Key Security

Along with the Active Directory, the next release of Windows NT Server will implement a distributed security model. This distributed security model is based on the MIT Kerberos authentication protocol. Kerberos authentication is used for distributed security within a tree, and accommodates both public and private key security using the same Access Control List (ACL) support model of the underlying Windows NT operating system. The Active Directory is the store for the security system, including user accounts, groups, and domains. It replaces the registry account database and is a trusted component within the Local Security Authority (LSA).

A single sign-on to the Windows NT domain tree allows user access to resources anywhere in the corporate network. Easy-to-use administrator tools for security policy and account management reduce the cost of deploying the Windows NT operating system. Windows NT also provides a foundation for integrated security for the Microsoft BackOffice® family of products, including Microsoft Exchange, Microsoft SQL Server™, Microsoft SNA Server, and Microsoft Systems Management Server.

The MIT Kerberos V5 authentication protocol is supported with extensions for public key-based authentication in addition to password-based (secret key) authentication.

Public Key Security

The Active Directory also supports the use of X.509 v3 Public Key Certificates for granting access to resources for subjects (for example, users) that do not have Kerberos credentials. This type of user is most often someone from outside an organization who needs access to resources within the organization. For example, an aerospace firm may hire subcontractors who need access to specifications, plans, and so forth. The Active Directory allows X.509 v3 certificates issued by a trusted authority to be mapped onto Windows NT security groups. Thus, a non-Windows NT user with a certificate can be granted access to resources in the same way as a user with Kerberos credentials.

Drag-and-Drop Administration

The Active Directory provides intuitive and powerful administration tools. Objects can be hierarchically organized so that they can model large organizations. And the graphical user interface delivers one of the most requested administrative tools—a drag-and-drop control console. This console has a graphical user interface that provides an object-view of

administration. For example, to do pruning and grafting, the administrator would grab the top of the merge-from tree, and then drag it to the target domain. A dialog box asks the administrator to confirm the action. Of course, the administrator must have rights in the merge-from tree to merge it with another tree, and in the merge-to domain to bring new trees into it.

Scripting and OLE Automation

Anything that can be done through a UI should be able to be done programmatically or from a script. To allow an administrator to write command procedures, the Active Directory provides full support for OLE automation and scripting. This makes it possible to add, change, move, copy, and perform other administrative functions by scripted manipulation using Active Directory, and a scripting language such as Visual Basic, Java™, or others.

A critical need for customers who have installed Windows NT Server versions 3.5*x* or 4.0 is backward compatibility. The Active Directory was designed from the start with backward compatibility built-in. The Active Directory provides complete emulation of the Windows NT 3.5*x* and 4.0 directory services; administrative tools and applications written to the Win32® API will continue to work unmodified in Active Directory environments. A next generation Windows NT Domain Controller installed in a Windows NT 3.5*x* or 4.0 Domain looks and acts *exactly* like a Windows NT 4.0 Domain Controller. This means that an investment in existing Windows NT network infrastructure and applications is protected. Customers can deploy Windows NT Server 4.0 today with complete confidence that their investment will support a smooth migration to the Active Directory.

Easy Migration from Windows NT 3.5*1x* and 4.0

To provide smooth and trouble-free migration, the Active Directory is designed to operate in a mixed environment. A mixed domain, with both next generation and “down-level” Windows NT 3.5*1x* and 4.0 domain controllers, works and acts just like a Windows NT 4.0 domain.

The migration process from down-level servers to the Active Directory can take place one domain controller at a time. Once thea primary domain controller in the Windows NT 4.0*x* domain has been upgraded, the domain can be joined to a Tree. The next section describes one example of how this migration process could work.

Migrating from Windows NT 4.0 to the Active Directory

This example uses a simple Windows NT 4.x domain with three domain controllers: one Primary Domain Controller (PDC) and two

Backup Domain Controllers (BDCs). Figure 7 shows the initial configuration.

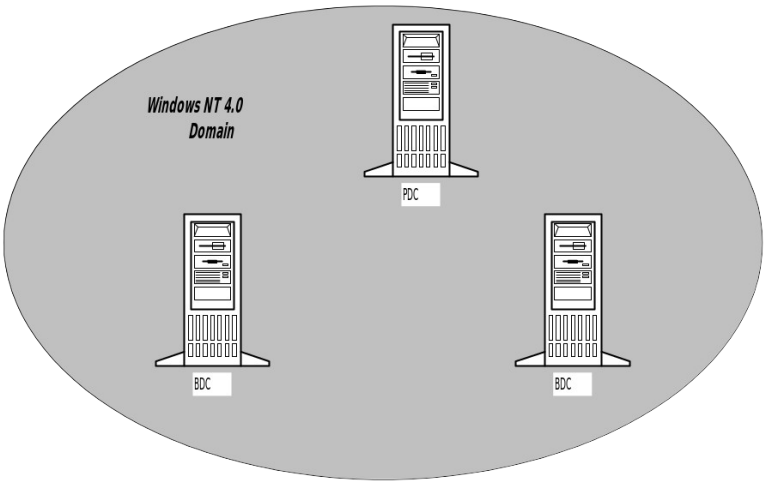


Figure 7. A simple Windows NT 4.x domain with a single Primary Domain Controller (PDC) and two Backup Domain Controllers (BDCs).

To begin the migration, you first upgrade the Primary Domain Controller (PDC) to Windows NT 5.0 and Active Directory.

The new DC/PDC populates the Active Directory from the Windows NT 4.0 domain directory during the upgrade (see Figure 8).

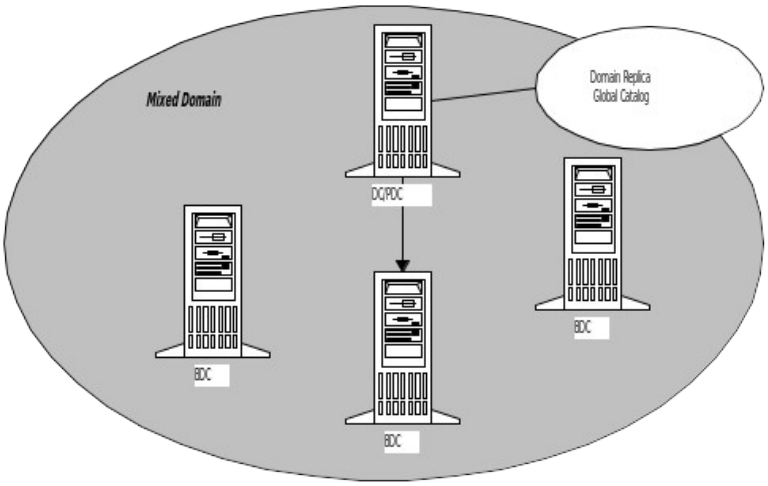


Figure 8. Mixed Domain

When you install Windows NT 5.0 at the PDC, you make the Active Directory the master copy of the domain directory. At this time, you can use Windows NT 5.0 graphical tools to perform system administration and account maintenance for the domain. The Windows NT 3.5*x* and 4.0 BDCs and the client systems in the domain are unaware of this change and continue to operate normally.

Your final migration step is to upgrade each BDC to Windows NT 5.0. As each BDC is converted, it becomes a peer of the PDC. Windows NT 4.0 replication is replaced by multimaster Windows NT 5.0 replication (see Figure 9).

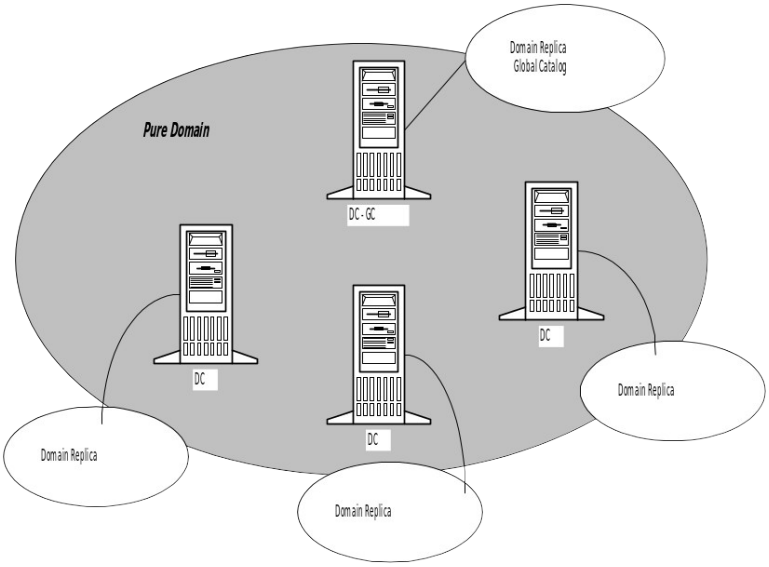


Figure 9. Pure Domain—the former BDCs are now peers of the original Windows NT 5.0 DC.

When all BDCs have been upgraded, the mixed domain becomes a pure Active Directory/Windows NT 5.0 domain. Down-level client systems will continue to see the domain as a Windows NT 3.5*x* or 4.0 domain. Next-generation clients will see the domain as a next generation domain and will be able to make full use of the Active Directory capabilities.

Easy Migration from the Microsoft Exchange Directory

Many organizations today are deploying Microsoft Exchange to provide their messaging and groupware infrastructure. These organizations will have a significant investment in the Microsoft Exchange directory when the Active Directory is released. Future releases of Microsoft Exchange will use the Active Directory, eliminating the need for a separate Microsoft Exchange directory. This release of Microsoft Exchange will provide a directory migration tool to transparently migrate the contents of the Microsoft Exchange directory to the Active Directory.

This approach helps protect customer investment in Microsoft Exchange data and organizational structure and provides a smooth migration path from the Microsoft Exchange directory to the Active Directory.

Microsoft is committed to enhancing its current Windows NT Server directory services in the next release of Windows NT Server. With this release, Microsoft will offer the Active Directory—standards-based directory services specifically designed to satisfy the needs of customers operating in a distributed computing environment.

Microsoft’s Active Directory gives enterprises the interoperability they need to unify and manage the multiple name spaces that now exist in

the heterogeneous software and hardware environments of enterprise networks. By combining the best of the DNS and X.500 naming standards, LDAP, other key protocols and a rich set of APIs, the Active Directory allows a single point of administration for all resources, including: files, peripheral devices, host connections, databases, Web access, users, arbitrary other objects, services, and network resources. The Active Directory supports a hierarchical name space for user, group and machine account information, and can subsume and manage other directories to provide a general-purpose directory service that can reduce the administrative burdens and costs associated with maintaining multiple name spaces.

The Active Directory, with its powerful combination of open standards, drag-and-drop administration, global catalog, extensibility, multimaster replication, distributed security, scalability, and complete backwards compatibility with Windows NT 3.5*x* and 4.0, makes the directory the ideal platform for administering the heterogeneous network of today, and the basis for the unified distributed computing environment of tomorrow.

For More Information

For the latest information on Windows NT Server, check out our World Wide Web site at http://www.microsoft.com/backoffice or the Windows NT Server Forum on the Microsoft Network (GO WORD: MSNTS). To access information via the World Wide Web, go to http://www.microsoft.com and select Microsoft BackOffice.